DL Assignment 2

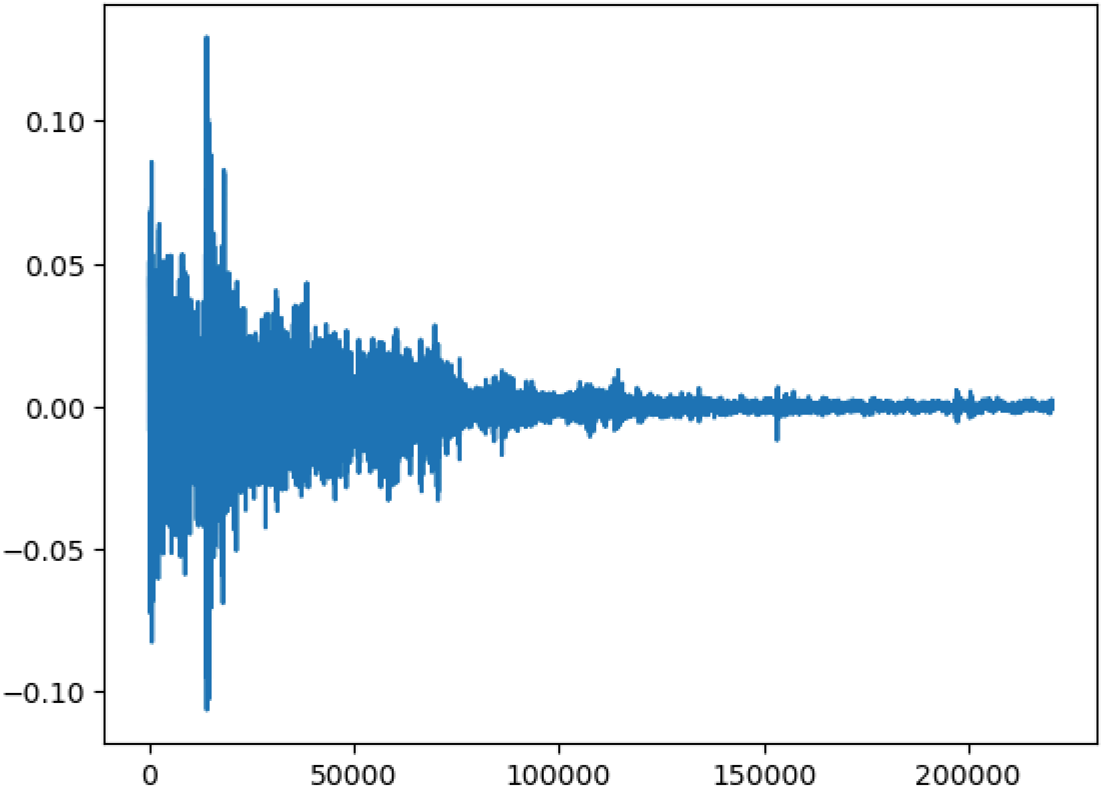
Alli Khadga Jyoth - M23CSA003

Link : M23CSA003.ipynb

Wandb Link: DL Assignment 2 Runs

Dataset

This is a dataset containing 400 data points of sounds generated by different things. Our aim is to classify these data points using CNN models and Transformer architecture and compare the performance.
The dataset is divided into 5 folds, the first of which is used for testing, and the remaining folds for 4-fold validation.



Example sample : Crackling_fire

Experiment Details

We define two models, a CNN architecture and a Transformer model, to classify the sound samples into 10 different classes. The individual implementation details are given in further sections. The training is done for 100 epochs.

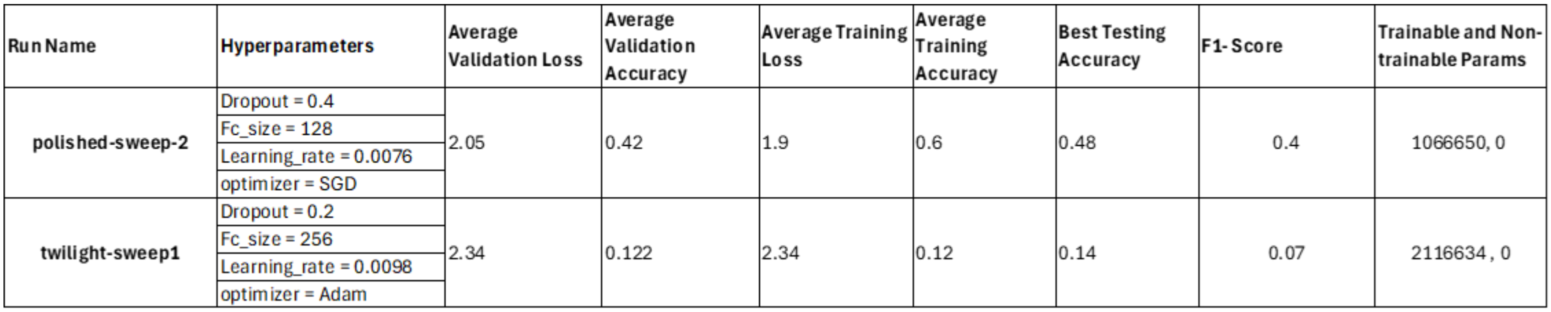
Hyperparameter tuning:

The model hyperparameters are trained using a random search, and the best hyperparameters are selected based on the "average validation accuracy" of the 4-fold validation. Then, the best test accuracy is reported as the best testing accuracy we observed during the 4-fold validation. The hyperparameter sweeps are reported differently for different models in their respective sections.

Architecture 1:

The CNN model contains 4 convolution layers, combined with max pooling and an adaptive-average pooling layer at the output layer. The model architecture is given below. The hyperparameter sweep configuration is reported in table1, along with the best test accuracy. From the table it is shown that the best hyperparameters are achieved from the sweep: “polished-sweep-2”. The plots for 4-fold validation are given in the figures below.
Run Link : DL_A2_Arch1 Workspace – Weights & Biases (wandb.ai)

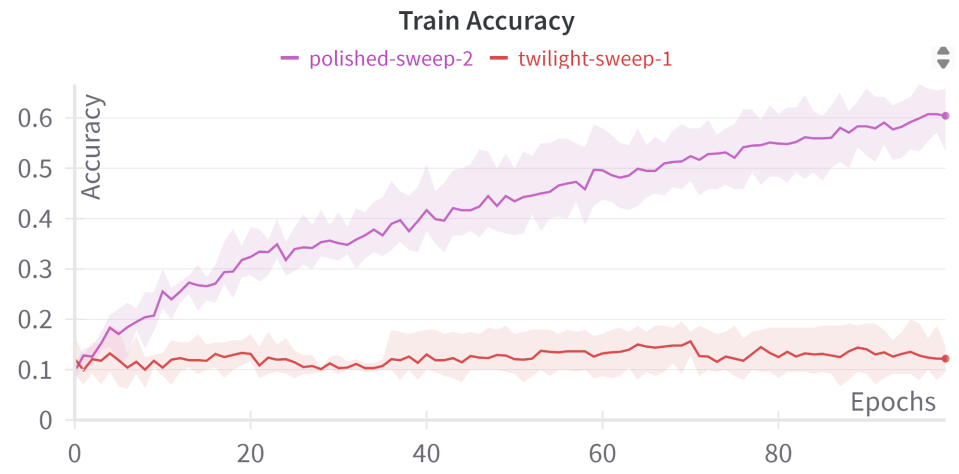
Table1: Hyperparameter sweep for CNN Architecture

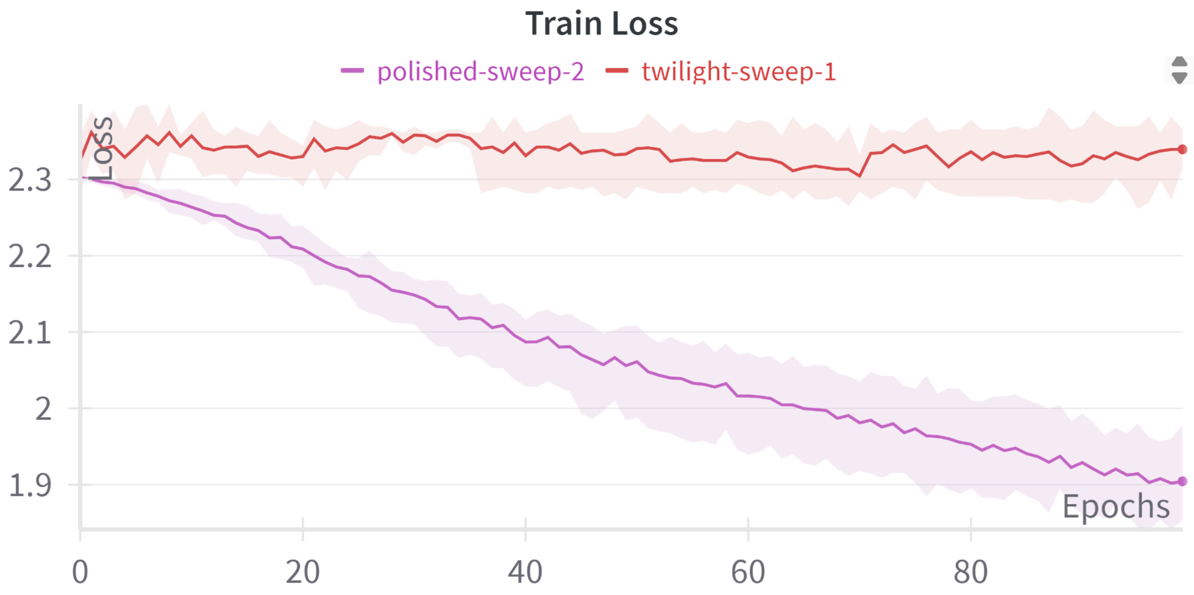


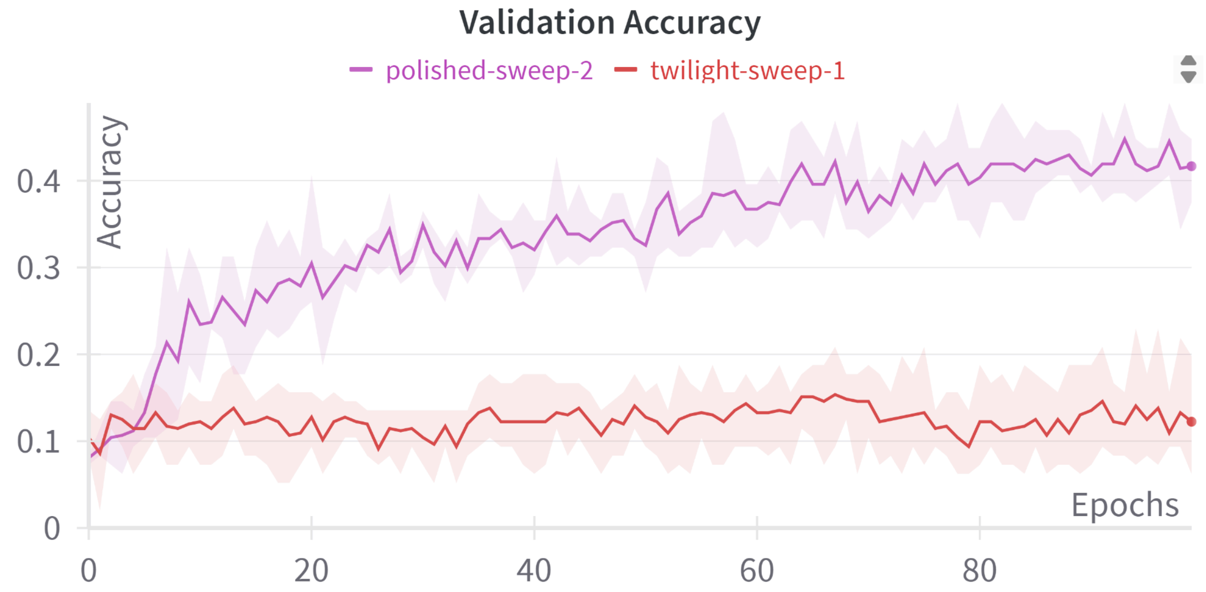
Run Name	Hyperparameters	Average Validation Loss	Average Validation Accuracy	Average Training Loss	Average Training Accuracy	Best Testing Accuracy	F1- Score	Trainable and Non-trainable Params
polished-sweep-2	Dropout = 0.4 Fc_size = 128 Learning_rate = 0.0076 optimizer = SGD	2.05	0.42	1.9	0.6	0.48	0.4	1066650, 0
twilight-sweep1	Dropout = 0.2 Fc_size = 256 Learning_rate = 0.0098 optimizer = Adam	2.34	0.122	2.34	0.12	0.14	0.07	2116634 , 0

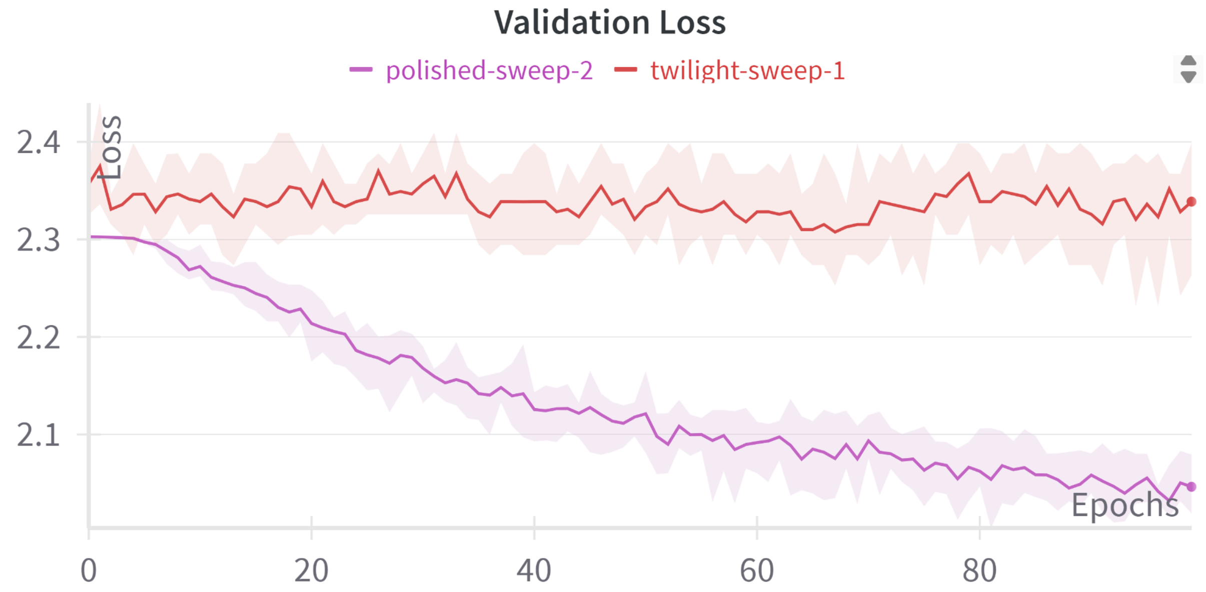
CNN Architecture:

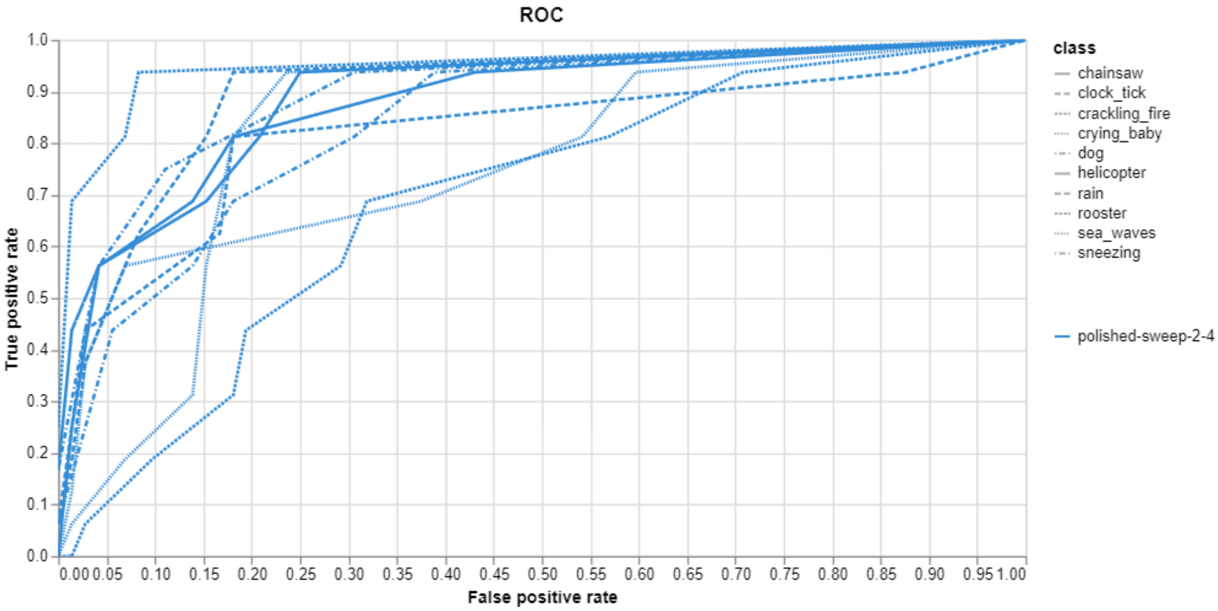
```
-------------------------------
        Layer (type)
===============================
            Conv1d-1
            Conv1d-2
       BatchNorm1d-3
            Conv1d-4
            Conv1d-5
       BatchNorm1d-6
            Linear-7
            Linear-8
===============================
```

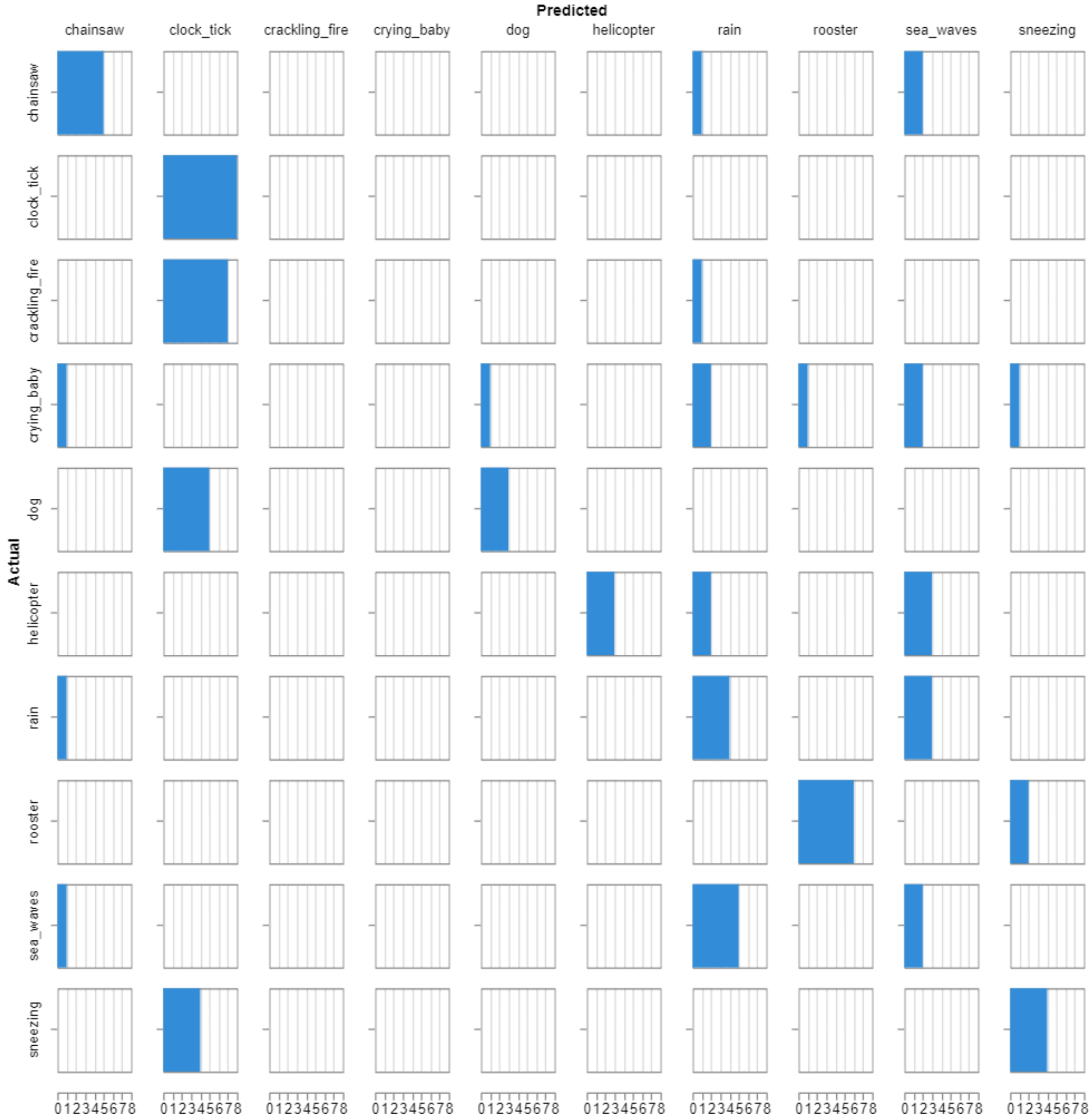





Validation Accuracy


Validation Loss


ROC
True positive rate
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
False positive rate
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0.50
0.55
0.60
0.65
0.70
0.75
0.80
0.85
0.90
0.95
1.00
class
chainsaw
clock_tick
crackling_fire
crying_baby
dog
helicopter
rain
rooster
sea_waves
sneezing
polished-sweep-2-4


Predicted
chainsaw
clock_tick
crackling_fire
crying_baby
dog
helicopter
rain
rooster
sea_waves
sneezing
Actual
chainsaw
clock_tick
crackling_fire
crying_baby
dog
helicopter
rain
rooster
sea_waves
sneezing
012345678
012345678
012345678
012345678
012345678
012345678
012345678
012345678
012345678
012345678

Architecture 2:

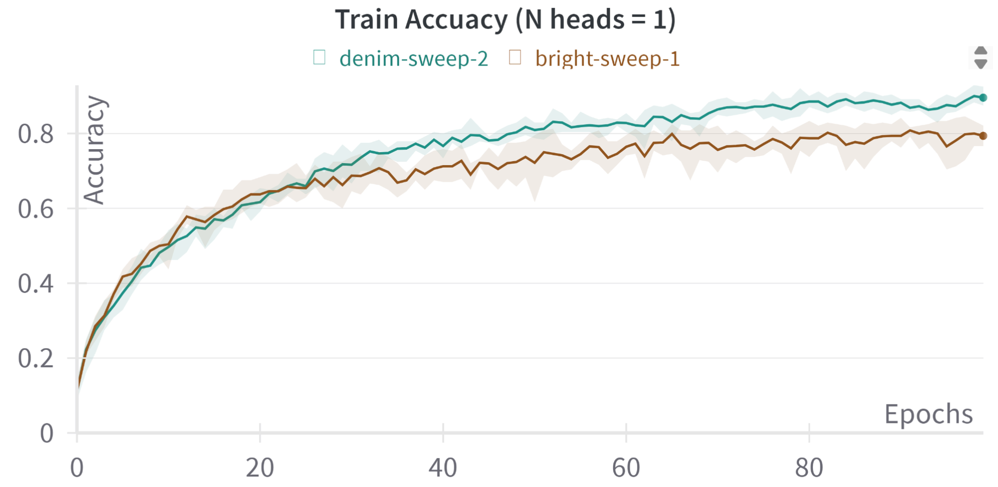
For this model we use a CNN feature extractor combined with a Transformer encoder to generate latent representations of the input which are stored in the ‹CLS› token to classify the input sound samples. The architecture is shown below. For this experiment we are required to report results with 1, 2, 4 number of heads in the multi-head attention block in the transformer encoder. The hyperparameter tuning and results are shown in the table2. With the accuracy, ROC, Confusion matrix following it. We get the best results for the sweep configuration "bright-sweep-1", which contains 1 heads combined with 2 encoder blocks and other hyperparameters.
Runs Link : DL_A2_Arch2_cls_final Workspace – Weights & Biases (wandb.ai)

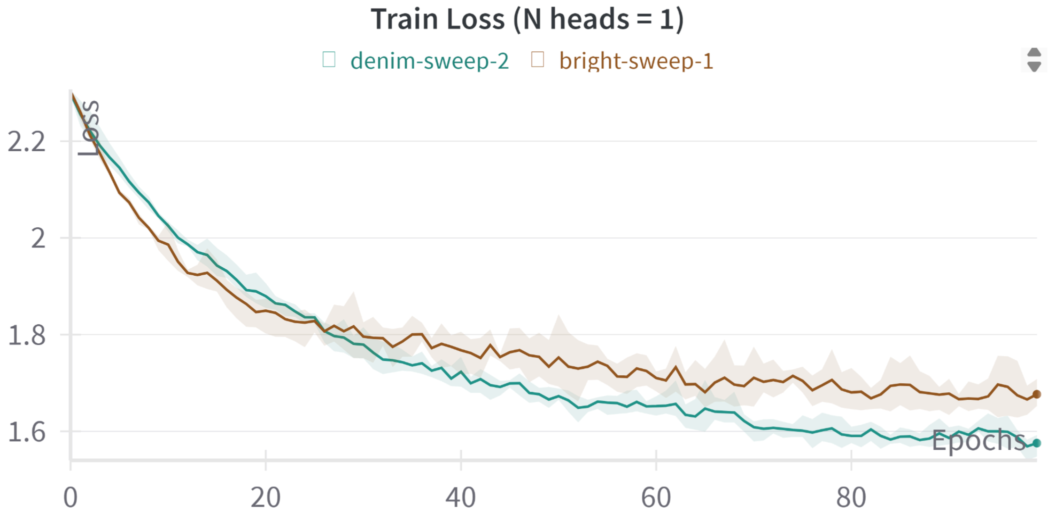
Table 2: Hyperparameter Sweep for Transformer Architecture

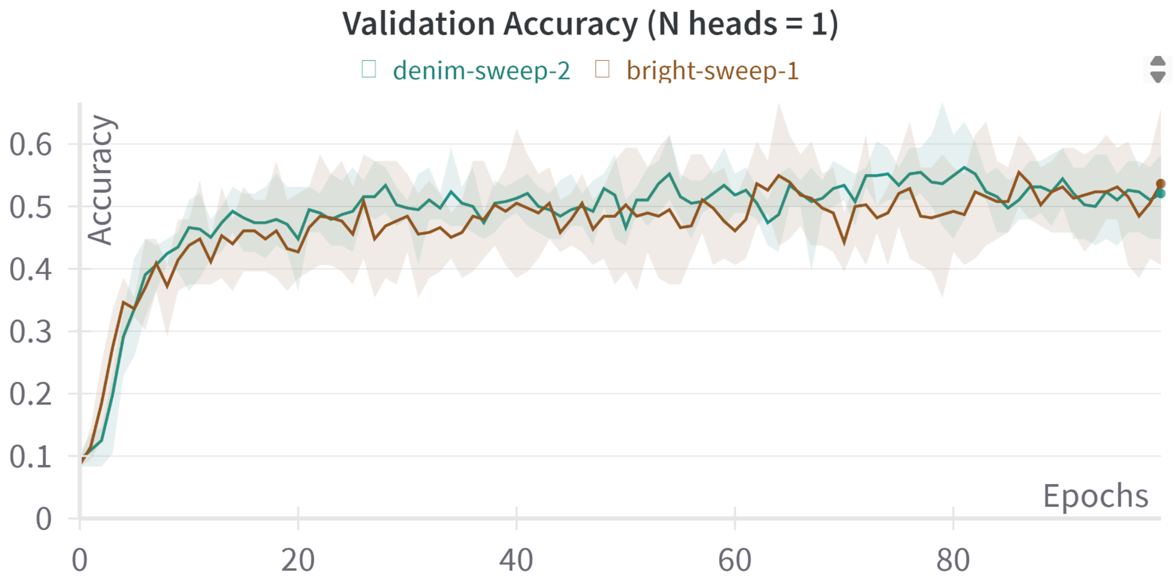
n_heads	Hyperprameters	Average Validation Loss	Average Validation Accuracy	Average Training Loss	Average Training Accuracy	Best Testing Accuracy	F1 - Score	Trainable and Non-trainable Params
1	name = **bright-sweep-1** dk = 64 dv = 32 features_length = 16 learning_rate = 0.0027 num_blocks = 2 optimizer = Adam	1.93	0.54	1.68	0.79	**0.56**	0.54	29658, 0
	name = **denim-sweep-2** dk = 64 dv = 32 features_length = 32 learning_rate = 0.0013 num_blocks = 3 optimizer = Adam	1.93	0.52	1.58	0.89	0.53	0.51	38370 , 0
2	name = **quiet-sweep-1** dk = 32 dv = 64 features_length = 16 learning_rate = 0.0062 num_blocks = 2 optimizer = Adam	2.21	0.23	2.19	0.23	0.3	0.19	73562 , 0
	name = **mild-sweep-2** dk = 64 dv = 64 features_length = 32 learning_rate = 0.0065 num_blocks = 3 optimizer = Adam	2.29	0.11	2.29	0.11	0.15	0.05	129386, 0
4	name = **deft-sweep-1** dk = 64 dv = 64 features_length = 16 learning_rate = 0.0018 num_blocks = 3 optimizer = SGD	2.24	0.25	2.25	0.24	0.27	0.16	227290 , 0
	name = **devout-sweep-2** dk = 32 dv = 64 features_length = 32 learning_rate = 0.0022 num_blocks = 3 optimizer = SGD	2.2	0.28	2.2	0.28	0.33	0.21	178922, 0

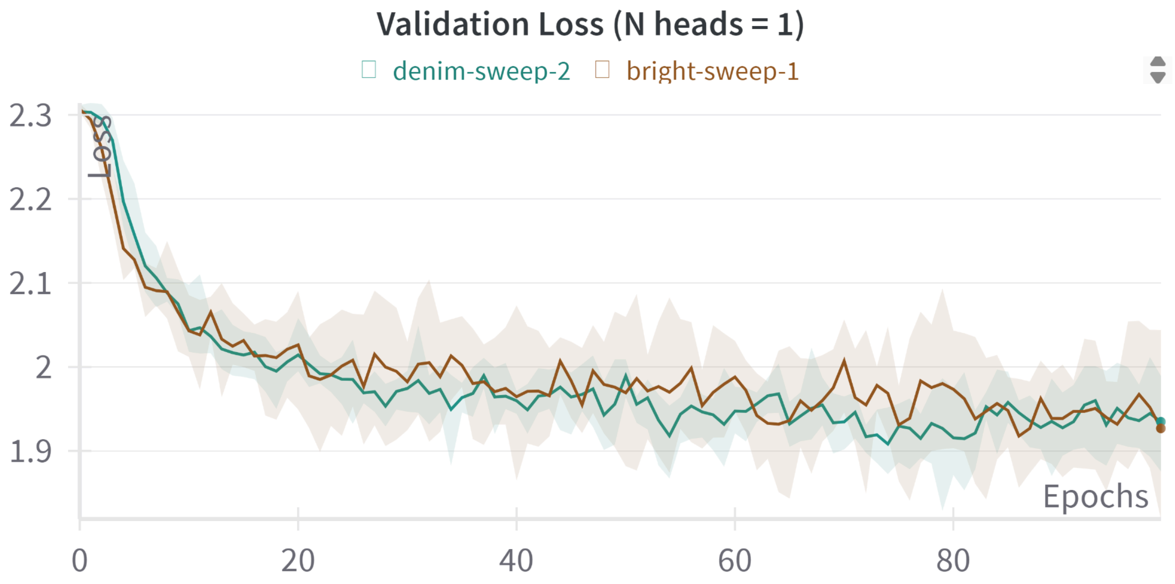
Transformer Architecture:

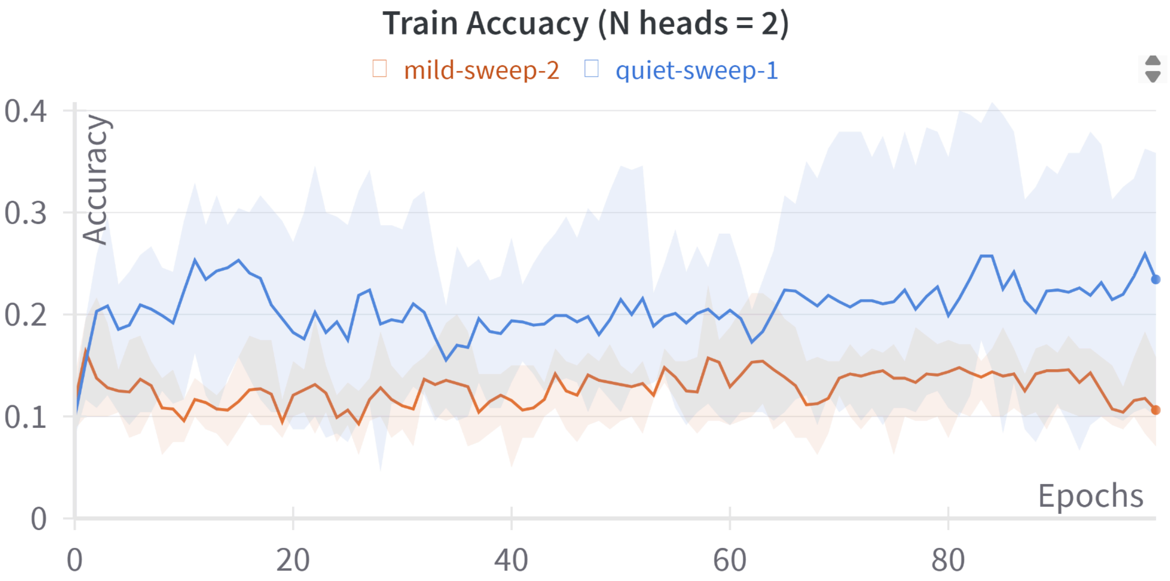
```
--------------------------
            Conv1d-2
       BatchNorm1d-3
            Conv1d-4
            Conv1d-5
       BatchNorm1d-6
      cnn_features-7
            Linear-8
            Linear-9
           Linear-10
           Linear-11
           Linear-12
           Linear-13
           Linear-14
multihead_attention-15
        LayerNorm-16
           Linear-17
        LayerNorm-18
    encoder_block-19
           Linear-20
           Linear-21
           Linear-22
           Linear-23
           Linear-24
           Linear-25
           Linear-26
multihead_attention-27
        LayerNorm-28
           Linear-29
        LayerNorm-30
    encoder_block-31
           Linear-32
           Linear-33
           Linear-34
           Linear-35
           Linear-36
           Linear-37
           Linear-38
multihead_attention-39
        LayerNorm-40
           Linear-41
        LayerNorm-42
    encoder_block-43
          encoder-44
           Linear-45
=============================
```

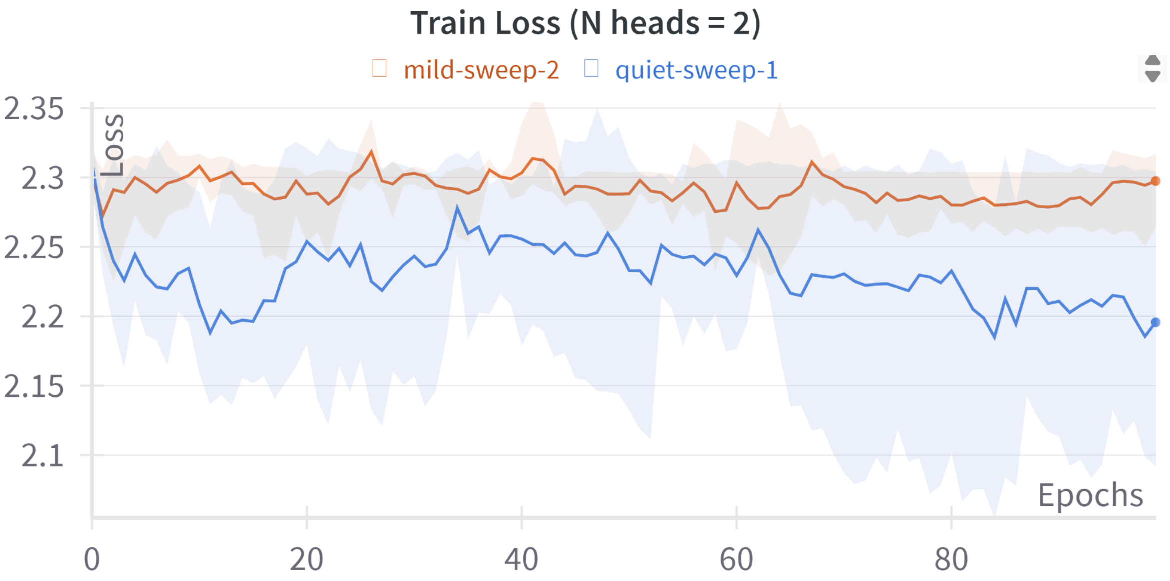


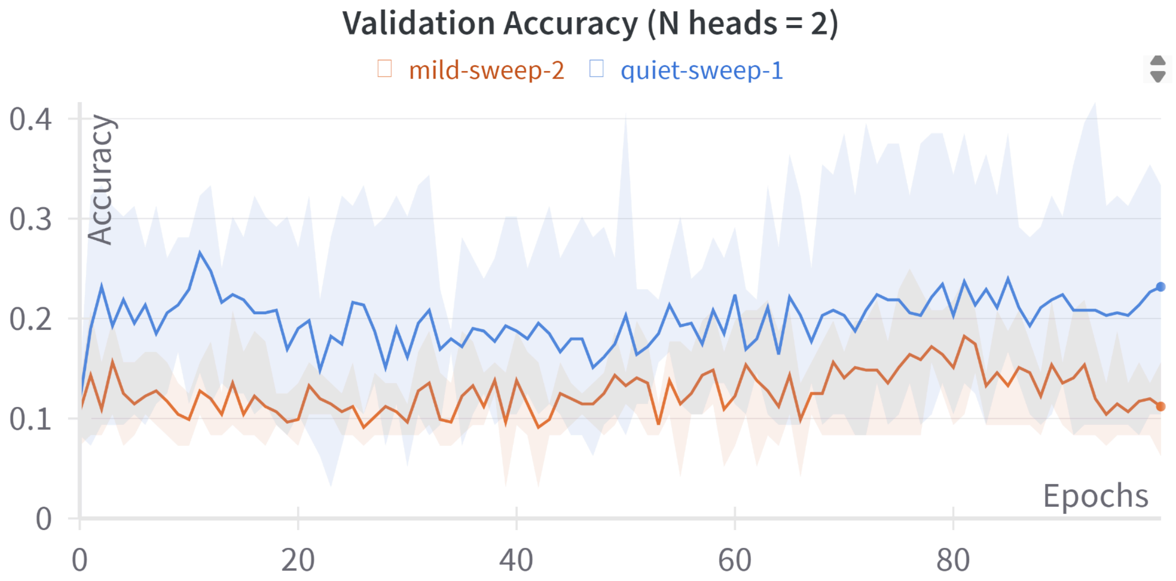


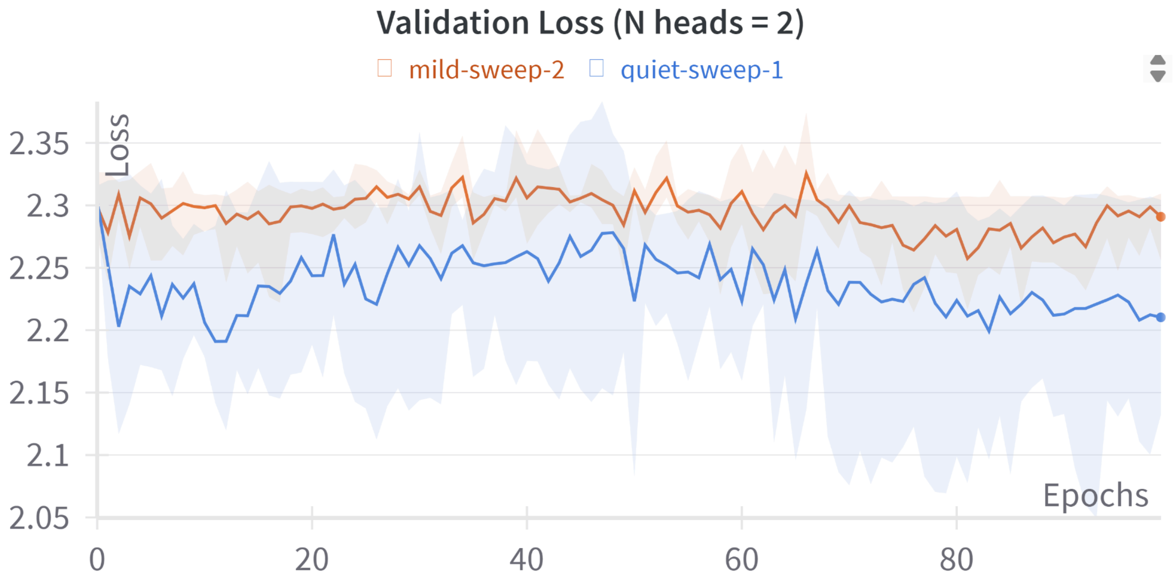

Validation Accuracy (N heads = 1)
□ denim-sweep-2
□ bright-sweep-1
Accuracy
0.6
0.5
0.4
0.3
0.2
0.1
0
Epochs
0
20
40
60
80

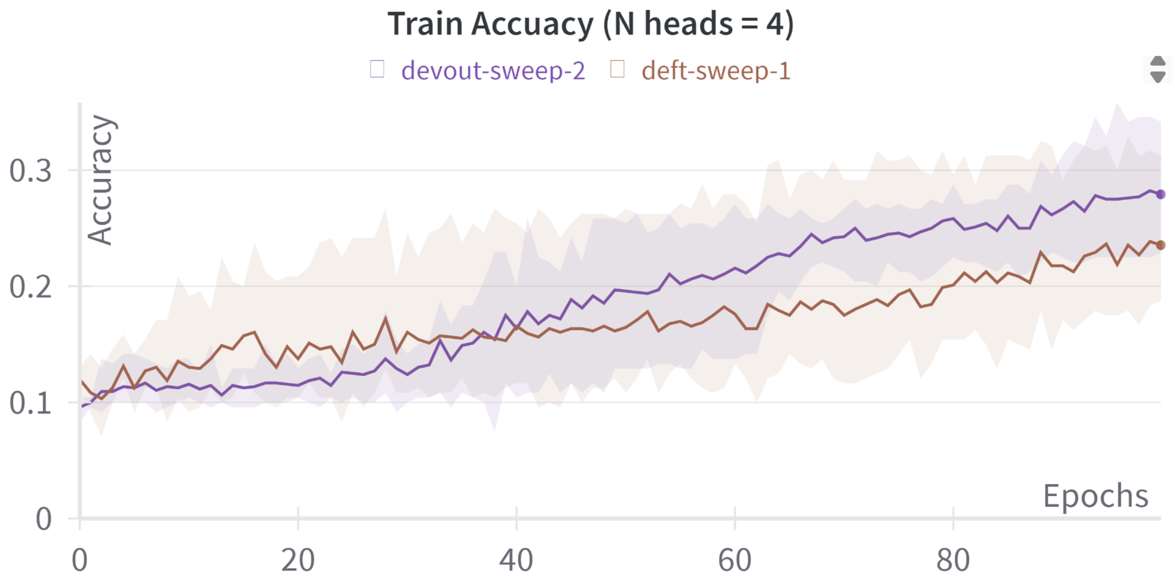

Validation Loss (N heads = 1)
□ denim-sweep-2
□ bright-sweep-1
Loss
2.3
2.2
2.1
2
1.9
Epochs
0
20
40
60
80

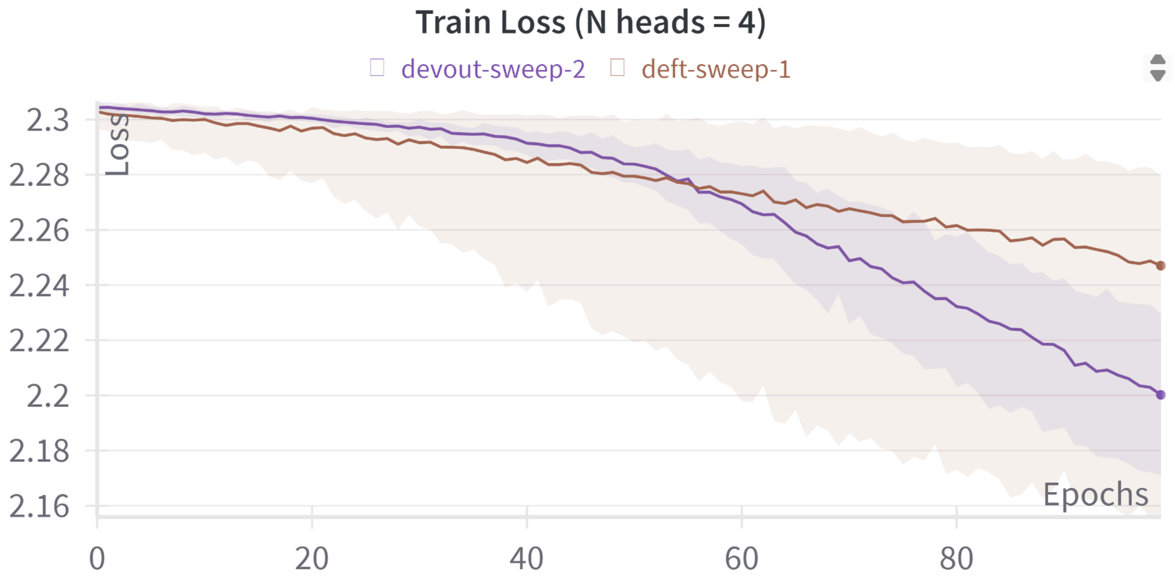

Train Accuacy (N heads = 2)
□ mild-sweep-2
□ quiet-sweep-1
Accuracy
0.4
0.3
0.2
0.1
0
Epochs
0
20
40
60
80

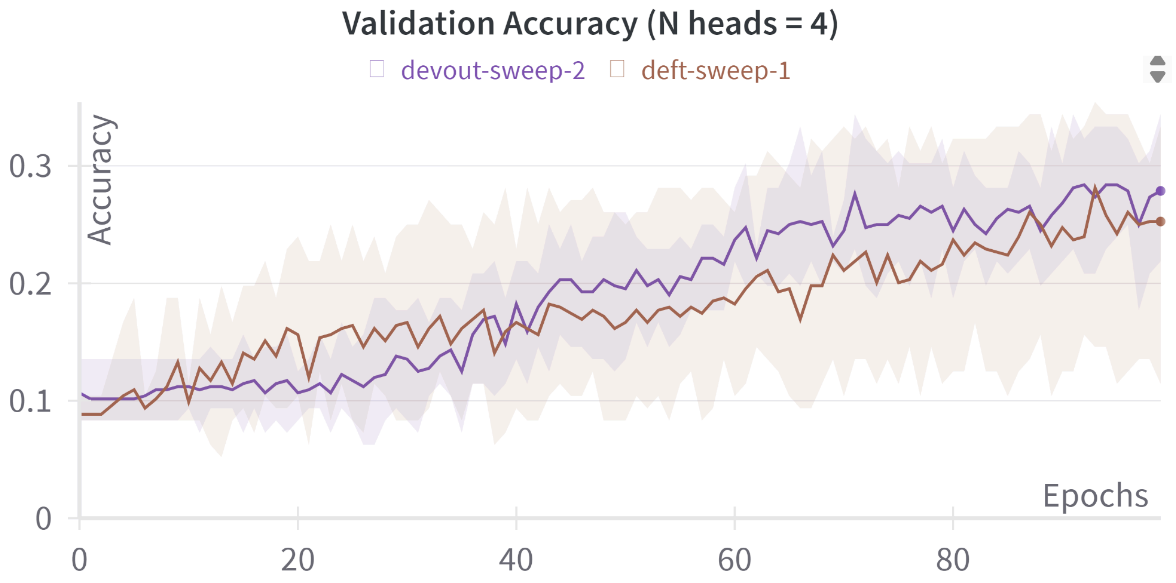

Train Loss (N heads = 2)
□ mild-sweep-2
□ quiet-sweep-1
Loss
2.35
2.3
2.25
2.2
2.15
2.1
Epochs
0
20
40
60
80

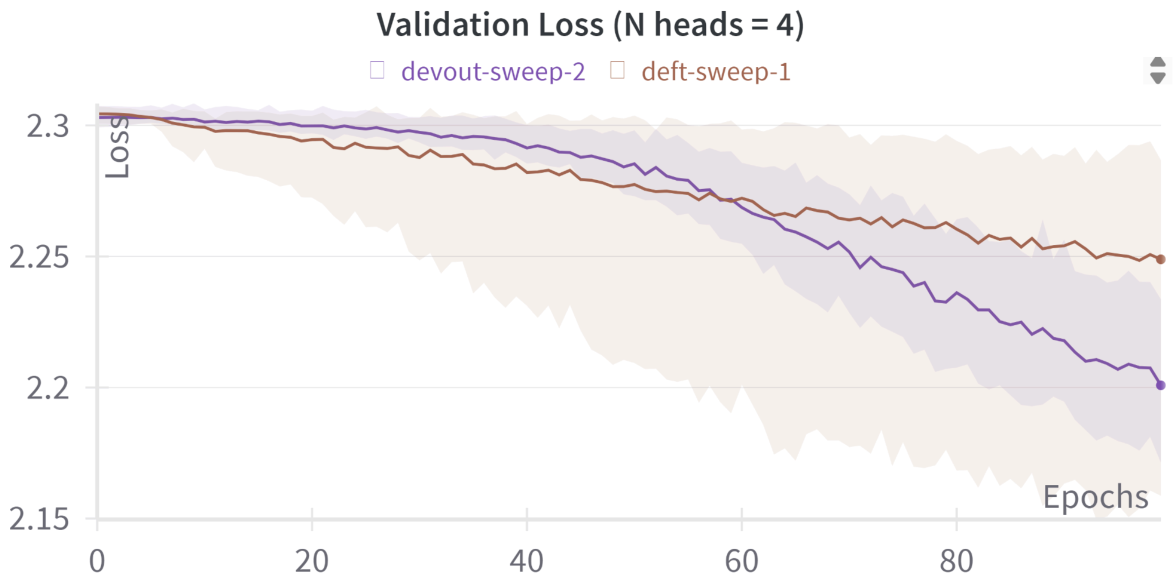

Validation Accuracy (N heads = 2)
□ mild-sweep-2
□ quiet-sweep-1
Accuracy
0.4
0.3
0.2
0.1
0
Epochs
0
20
40
60
80


Validation Loss (N heads = 2)
□ mild-sweep-2
□ quiet-sweep-1
Loss
2.35
2.3
2.25
2.2
2.15
2.1
2.05
Epochs
0
20
40
60
80

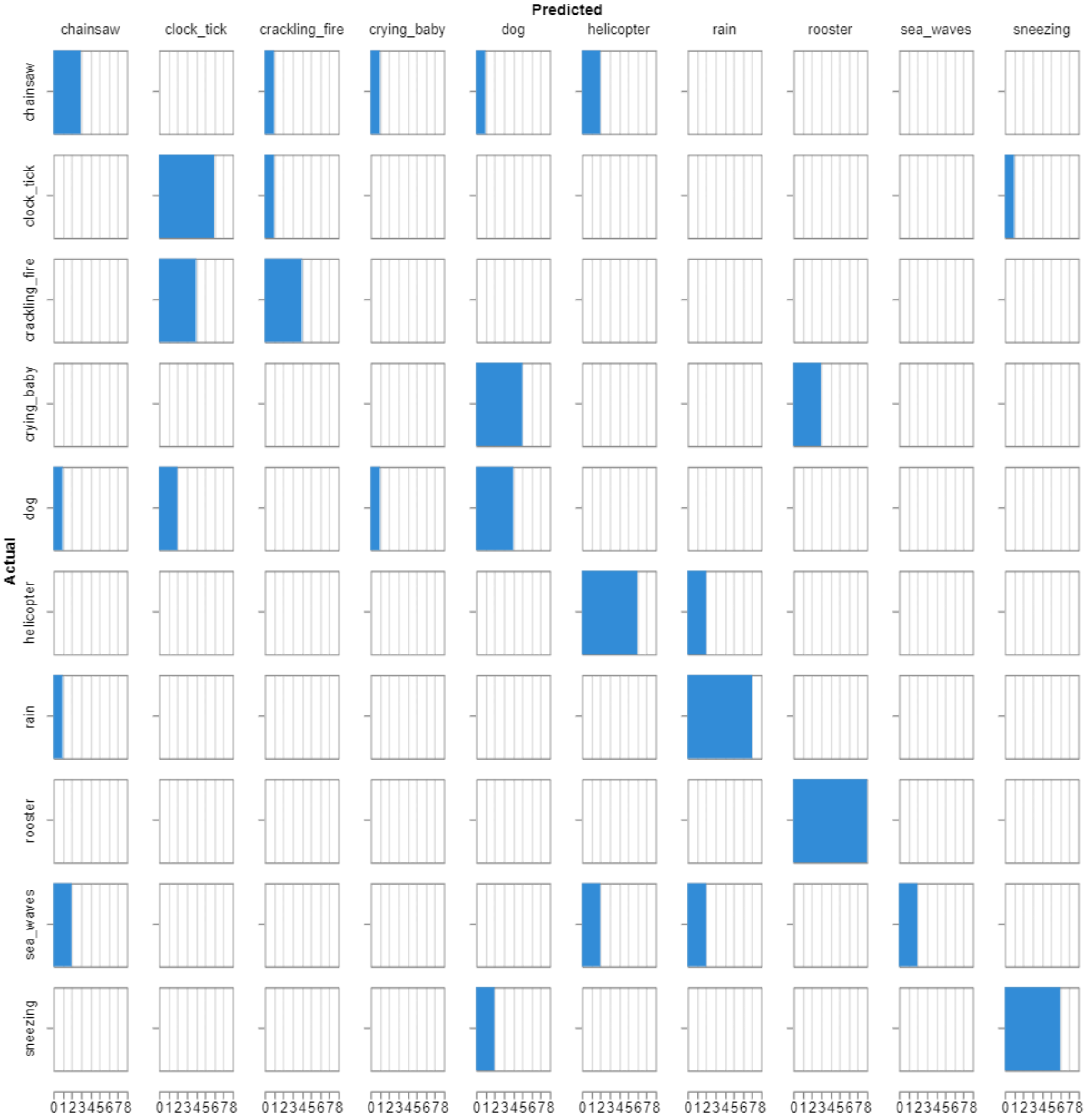

Train Accuacy (N heads = 4)
devout-sweep-2
deft-sweep-1
Accuracy
0.3
0.2
0.1
0
Epochs
0
20
40
60
80

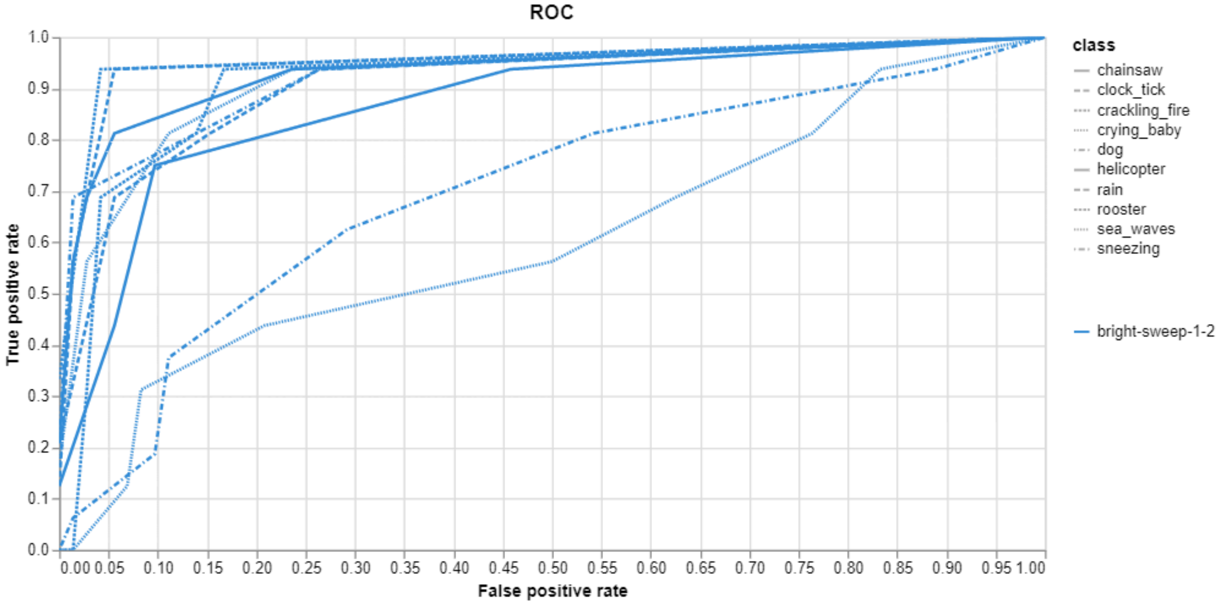

Train Loss (N heads = 4)
devout-sweep-2
deft-sweep-1
Loss
2.3
2.28
2.26
2.24
2.22
2.2
2.18
2.16
Epochs
0
20
40
60
80


Validation Accuracy (N heads = 4)
□ devout-sweep-2
□ deft-sweep-1
Accuracy
0.3
0.2
0.1
0
Epochs
0
20
40
60
80

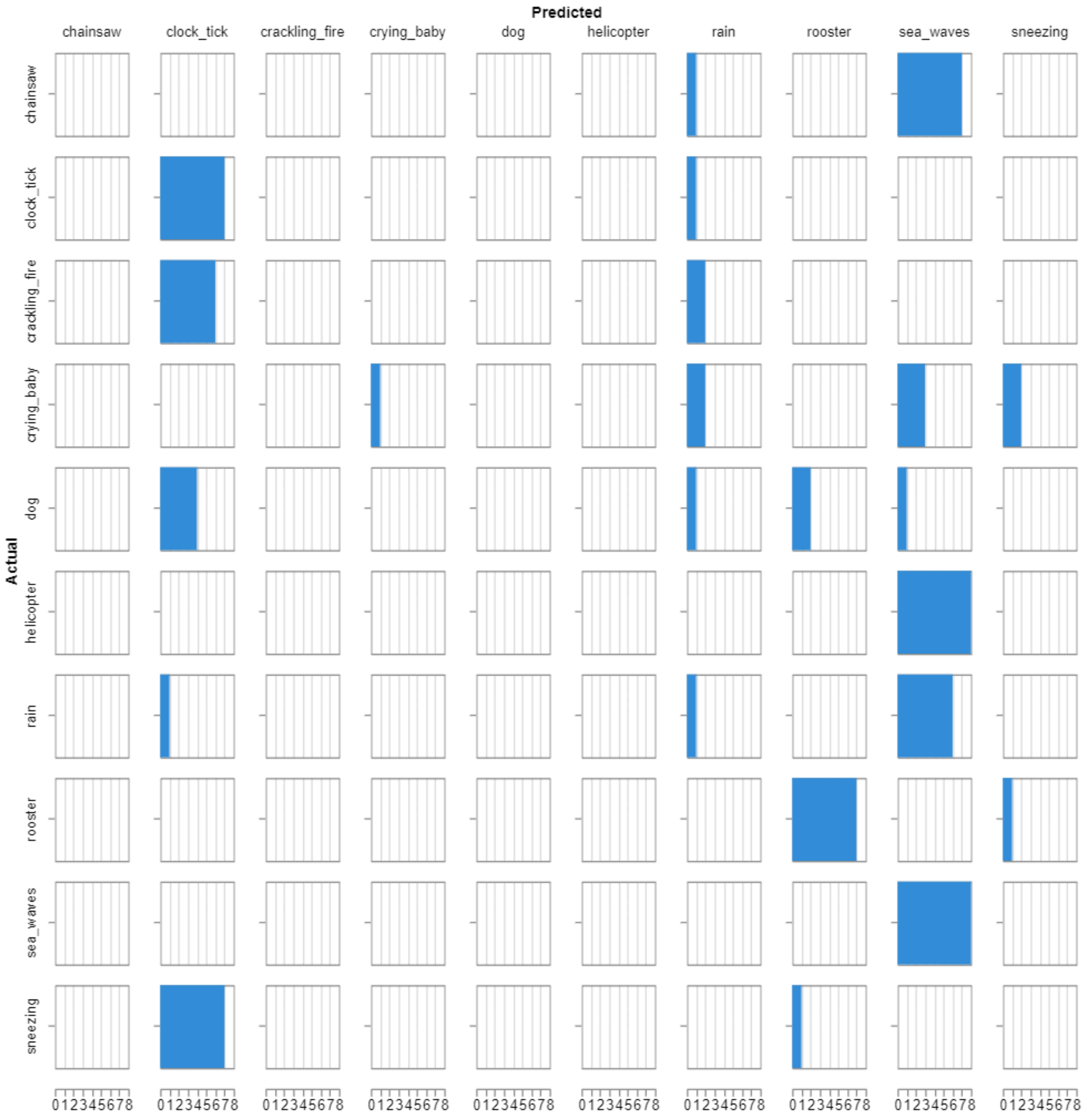

Validation Loss (N heads = 4)
□ devout-sweep-2
□ deft-sweep-1
Loss
2.3
2.25
2.2
2.15
Epochs
0
20
40
60
80

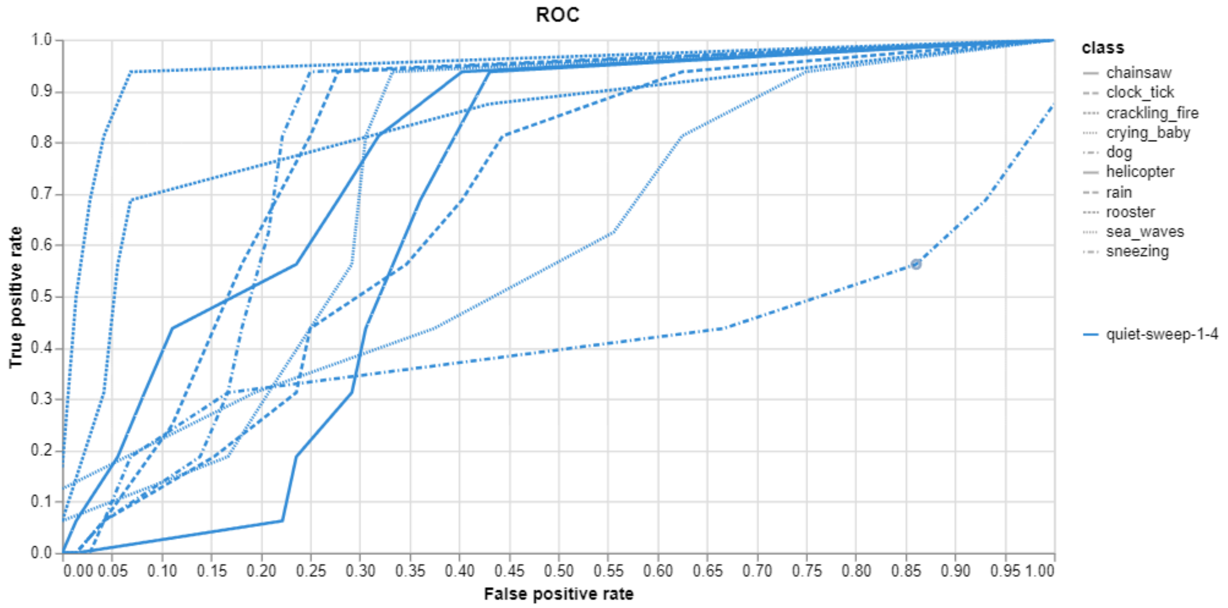
Confusion Matrix and AUC-ROC Curve : N Heads = 1



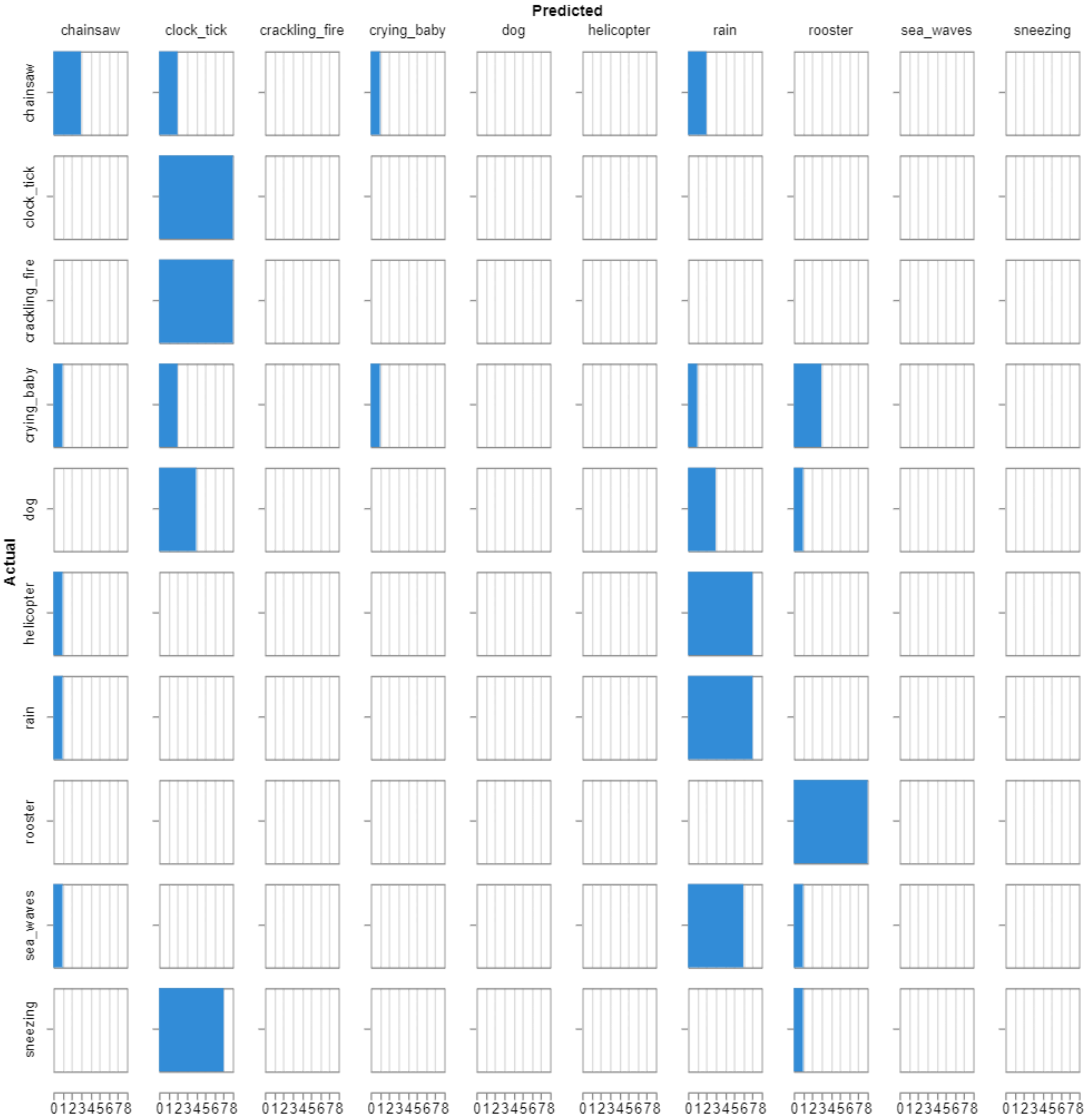


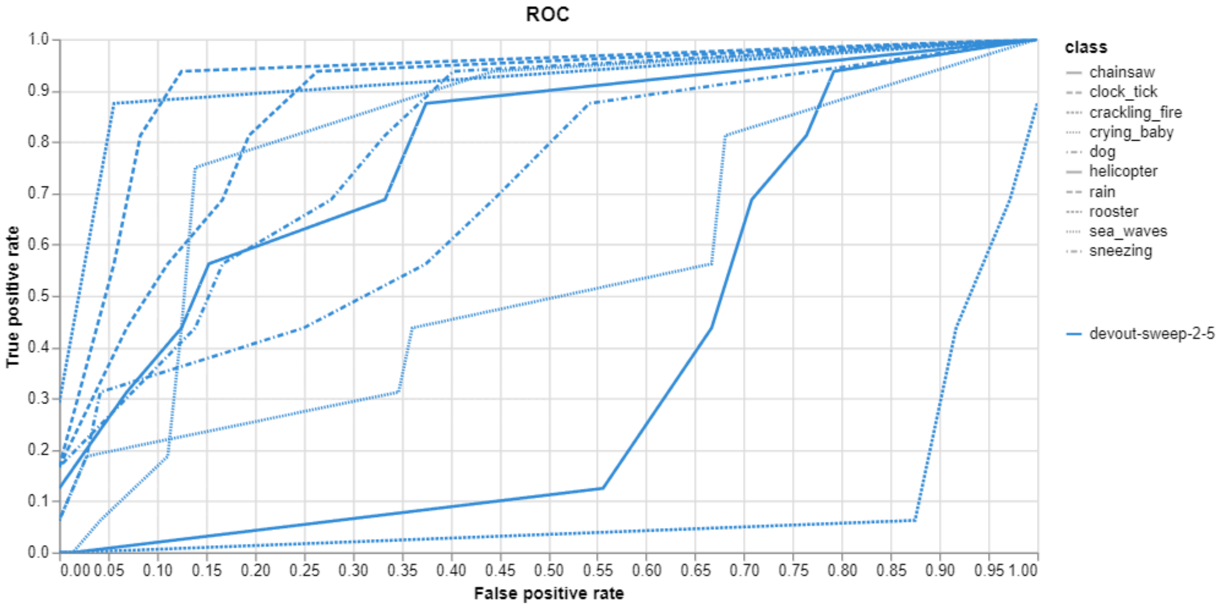
Confusion Matrix and AUC-ROC Curve : N Heads = 2





Confusion Matrix and AUC-ROC Curve : N Heads = 4





Results:

Comparing the results from both CNN architecture and Transformer architecture, 48% and 56% respectively, we see a difference of ~8%, this difference can be attributed to the attention mechanism of the Transformer model. Since we compare the number of parameters of CNN (~1.06M) vs the Transformer (~30K), we see a ~33x difference.

For CNN Architecture:

- The model "polished-sweep-2" with Dropout=0.4, Fc_size=128, Learning_rate=0.0076, and optimizer=SGD performed better in terms of Average Validation Accuracy, Average Training Accuracy, Best Testing Accuracy and F1-Score compared to "twilight-sweep1".
- Despite having more trainable parameters, "twilight-sweep1" performed poorly in all metrics compared to "polished-sweep-2".

For Transformer Architecture:

- The model "bright-sweep-1" with 1 head and Adam optimizer outperformed other models in terms of Average Validation Accuracy, Average Training Accuracy, Best Testing Accuracy and F1-Score.
- Increasing the number of heads from 1 to 4 did not necessarily lead to better performance.
- Models with SGD as the optimizer ("deft-sweep-1", "devout-sweep-2") had relatively lower performance metrics compared to those using Adam.
- The model with the lowest learning rate (0.0013) achieved higher training accuracy.